Developing a Dynamic Mapping to Manage Metadata Changes in Relational Sources

Abstract

In the Developer tool, you can develop a dynamic mapping that handles metadata changes in relational sources at run time. This article describes the steps to create a dynamic mapping for relational tables that can have metadata changes and to run the mapping with metadata changes. This article assumes that you are familiar with mapping and dynamic mapping concepts and know how to develop and run a mapping.

Supported Versions

- Informatica Big Data Management 10.0
- Informatica Data Quality 10.0
- Informatica Data Services 10.0

Table of Contents

Overview

A dynamic mapping is a mapping in which you can change sources, targets, and transformation logic at run time based on parameters and rules that you define. When your organization wants to manage frequent metadata changes in the data sources, develop a dynamic mapping that can get the metadata changes directly from the data sources at run time.

Use Case: Dynamic Mapping for Metadata Changes in Relational Sources

You are a developer for an organization that must aggregate total customer orders. The organization receives customer data and customer order data as two tables from different departments on a weekly basis. The departments often change the order of the columns or add new columns to the tables. You need to develop a dynamic mapping that can accommodate the changing source schema and aggregate the total customer orders.

Source Tables

CUSTOMER and ORDERS are the source tables for the Read transformations in the mapping.

The following table lists the columns and metadata for the CUSTOMER table with the C_CUSTKEY column as the primary key:

Name	Native Type	Precision	Scale
C_CUSTKEY	number(p,s)	38	0
C_NAME	varchar2	25	0
C_ADDRESS	varchar2	40	0
C_NATIONKEY	number(p,s)	38	0
C_PHONE	varchar2	15	0
C_ACCTBAL	number(p,s)	10	2
C_MKTSEGMENT	varchar2	10	0

The following table lists the columns and metadata for the ORDERS table:

Name	Native Type	Precision	Scale
O_ORDERKEY	number(p,s)	38	0
O_CUSTKEY	number(p,s)	38	0
O_ORDERSTATUS	varchar2	1	0
O_TOTALPRICE	number(p,s)	10	2
O_ORDERDATE	date	19	0
O_ORDERPRIORITY	varchar2	15	0
O_CLERK	varchar2	15	0
O_SHIPPRIOIRITY	number(p,s)	30	0

Target Table

CUSTOMERSUMMARY is the target table for the Write transformation in the mapping.

The following table lists the columns and metadata for the CUSTOMERSUMMARY table:

Name	Native Type	Precision	Scale
C_CUSTKEY	number(p,s)	38	0
C_NAME	varchar2	25	0
C_ADDRESS	varchar2	40	0
C_NATIONKEY	number(p,s)	38	0

Name	Native Type	Precision	Scale
C_PHONE	varchar2	15	0
C_ACCTBAL	number(p,s)	10	2
C_MKTSEGMENT	varchar2	10	0
C_TOTALAMOUNT	number(p,s)	10	2

Dynamic Mapping

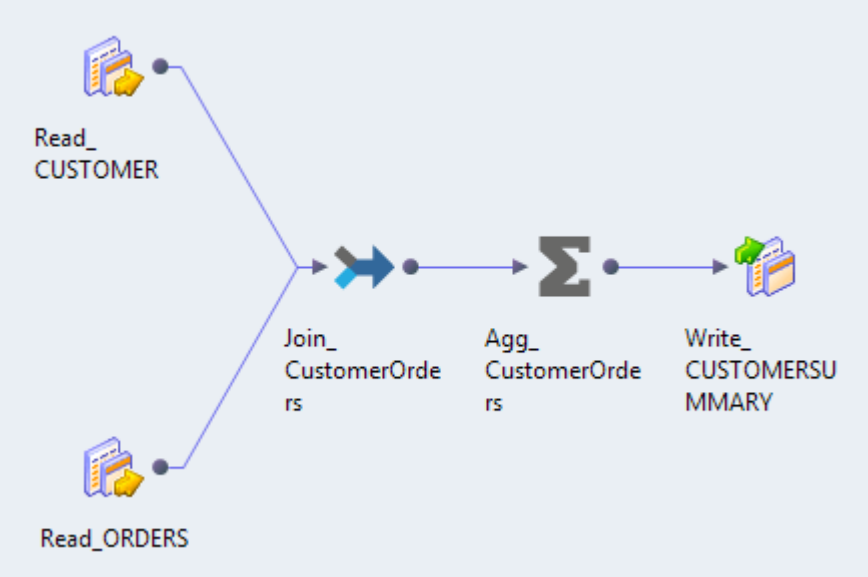
Create a mapping m_CustomerLoad and configure the following dynamic mapping functionality:

- Read transformations that can read from dynamic sources
- Dynamic ports in the downstream transformations that can pass new and changed columns
- Write transformation that can write to dynamic targets
- Run-time links that can connect ports to the Write transformation at run time

When you run the mapping, the Data Integration Service performs the following tasks:

1. Fetches the structure of the data objects and metadata changes in the source files.
2. Passes the new and changed columns to each transformation through dynamic ports.
3. Connects the new and changed ports to the Write transformation.
4. Writes the transformed data to the target.

The following image shows the objects in the mapping:



The mapping contains the following objects:

Read_CUSTOMER

Read transformation that represents the relational source CUSTOMER. The relational table contains a separate row for each customer.

Read_ORDERS

Read transformation that represents the relational source ORDERS. The relational table that contains a separate row for each customer order.

Join_CustomerOrders

Joiner transformation that joins the CUSTOMER and ORDERS sources.

Agg_CustomerOrders

Aggregator transformation that aggregates the total customer orders.

Write_CUSTOMERSUMMARY

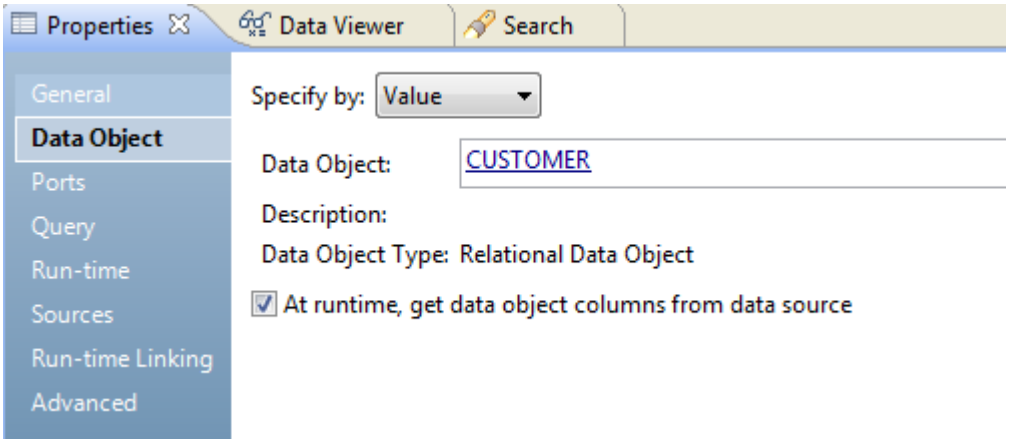
Write transformation that represents the relational target CUSTOMERSUMMARY. The relational table contains a column for the mapping to write the aggregated value for total orders grouped by customer.

Step 1. Configure the Read Transformations

Configure the Read transformations to fetch column and metadata changes directly from the dynamic sources at run time.

1. Add two Read transformations that represent the CUSTOMER and ORDERS relational data objects.
2. Configure the Read_CUSTOMER transformation to fetch column and metadata changes directly from the sources at run time.
   a. Select the Read_CUSTOMER transformation.
   b. In the **Properties** view, click the **Data Object** tab.
   c. Select **At run time, get data object columns from the data source**.

   The following image shows the Data Object tab settings of the Read_CUSTOMER transformation:



3. Configure the Read_ORDERS transformation to fetch column and metadata changes directly from the sources at run time.
   a. Select the Read_ORDERS transformation.
   b. In the **Properties** view, click the **Data Object** tab.
   c. Select **At run time, get data object columns from the data source**.

Step 2. Configure the Joiner Transformation

Add a Joiner transformation to the mapping and configure dynamic ports to receive any new and changed columns from the Read transformation. Define a join condition to join the two source tables CUSTOMER and ORDERS.

1. Add a Joiner transformation Join_CustomerOrders to the mapping.
2. Create dynamic ports in the Joiner transformation:
   a. From the Read_Customer transformation, drag the All Ports group to the Master group in the Joiner transformation.

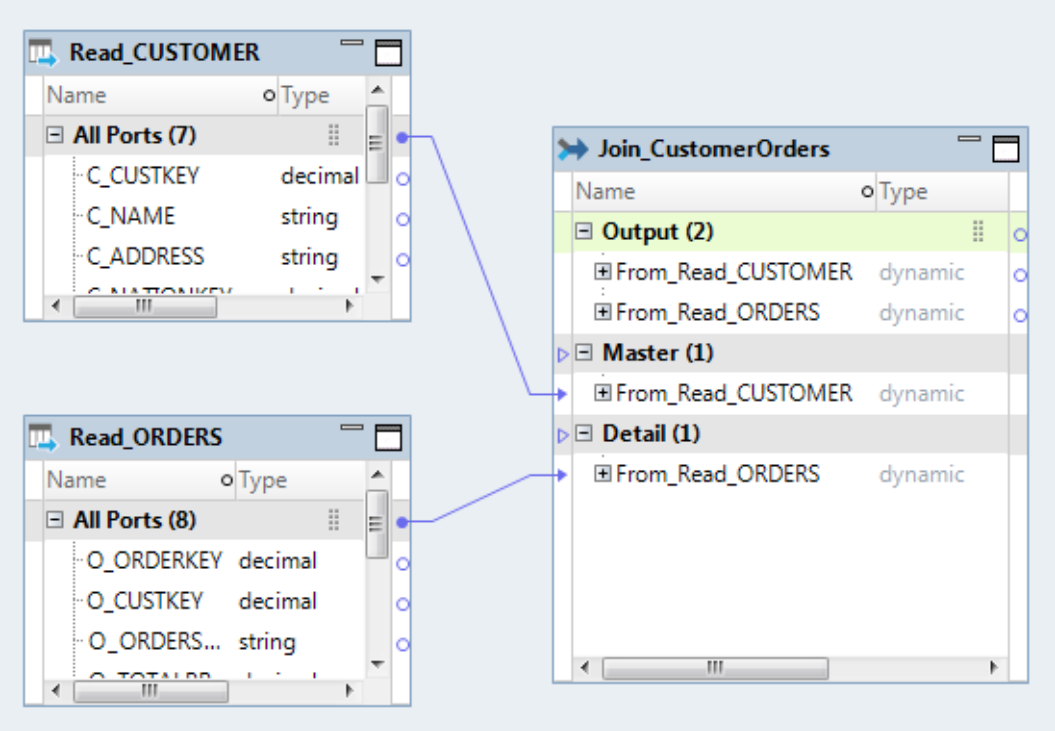
      The Developer tool creates a dynamic port From_Read_CUSTOMER in the Master group and the Output group.

   b. From the Read_Orders transformation, drag the All Ports group to the Detail group in the Joiner transformation.

      The Developer tool creates a dynamic port From_Read_ORDERS in the Detail group and the Output group.

   The dynamic ports include all ports from the corresponding Read transformations as generated ports.

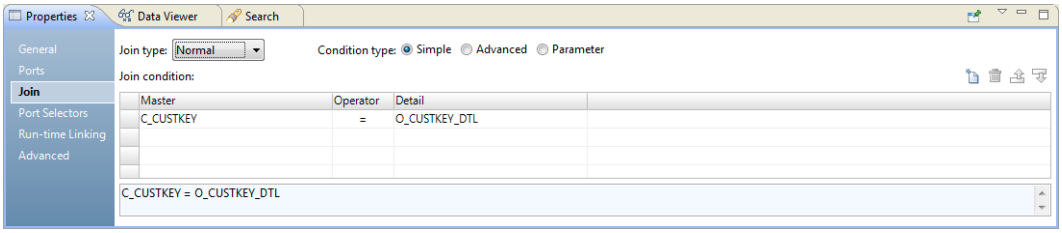
   The following image shows the All Ports groups from the Read transformations linked to the two dynamic ports in the Joiner transformation:



3. In the **Properties** view, click the **Join** tab.

4. Click the **New** button, and define the join condition as C_CUSTKEY = O_CUSTKEY_DTL.

   The following image shows the **Join** tab with the join condition defined:



Step 3. Configure the Aggregator Transformation

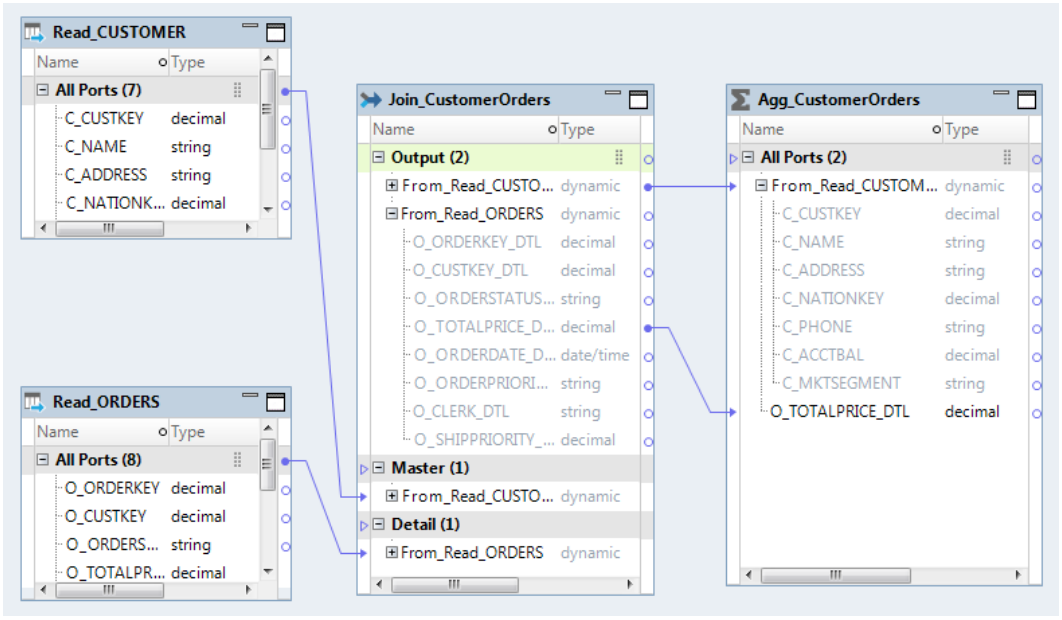
Add an Aggregator transformation to the mapping and configure dynamic ports to receive any new and changed columns from the Joiner transformation. Create an aggregate expression to calculate the total price of customer orders and group the aggregation by customer.

1. Add an Aggregator transformation Agg_CustomerOrders to the mapping.
2. Create dynamic ports in the Aggregator transformation:
   a. From the Output group in the Joiner transformation, drag the From_Read_CUSTOMER dynamic port to the Aggregator transformation.

      A dynamic port From_Read_CUSTOMER appears in the Aggregator transformation.

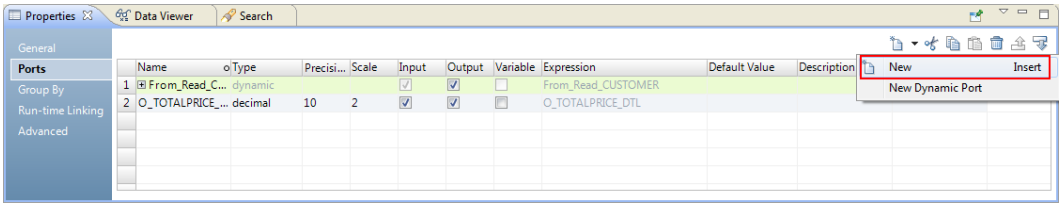
   b. From the From_Read_ORDERS dynamic port of the Output group in the Joiner transformation, drag the O_TOTALPRICE_DTL generated port to the Aggregator transformation.

   The following image shows the ports from the Joiner transformation linked to the Aggregator transformation:



3. In the **Properties** view, click the **Ports** tab.

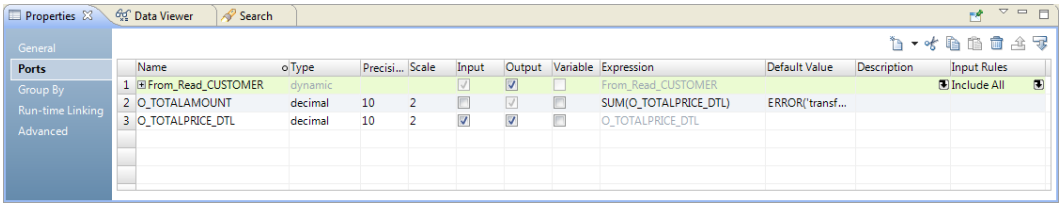
4. Click the **New** button to create a port to aggregate prices of the orders.



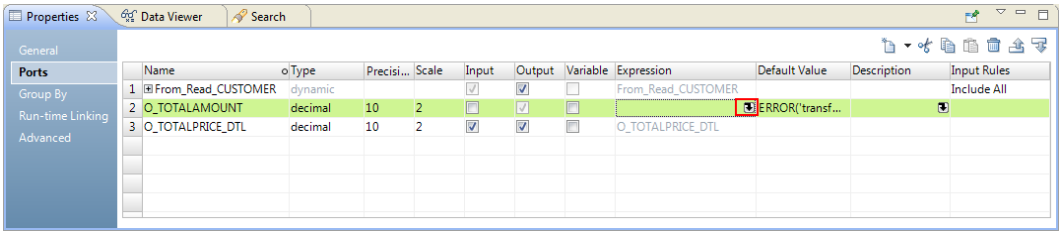
The Developer tool creates a new port called Field.

5. Select the new port, and change the column values as follows:
   - Name: O_TOTALAMOUNT
   - Type: decimal
   - Precision: 10
   - Scale: 2
   - Input: Clear the selection to make this port an output-only port.

   The following image shows the ports in the Aggregator transformation:

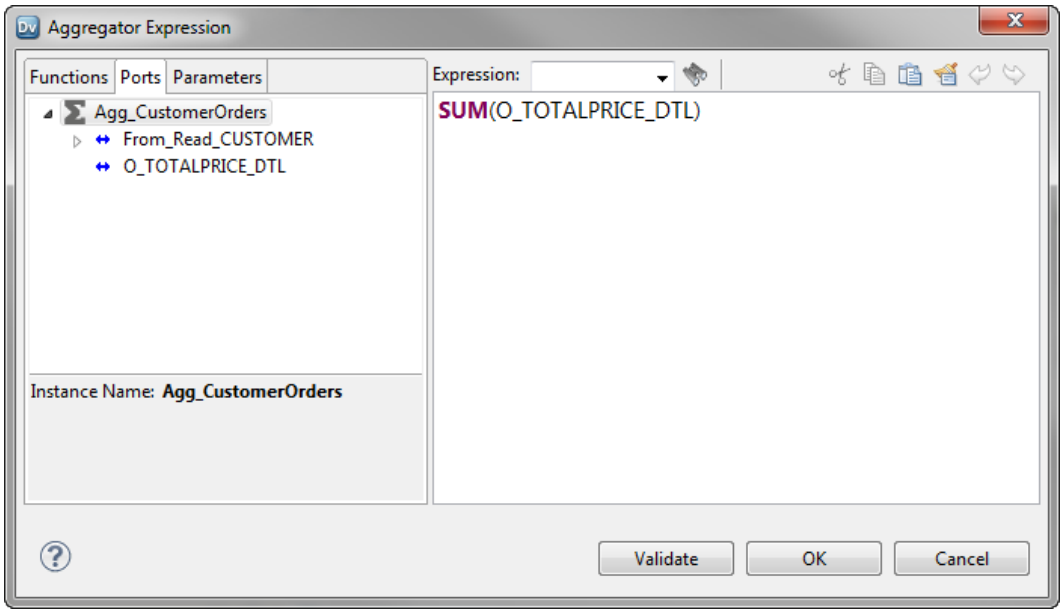


6. In the Expression column for the O_TOTALAMOUNT port, click the **Open** button.



The **Aggregator Expression** window appears.

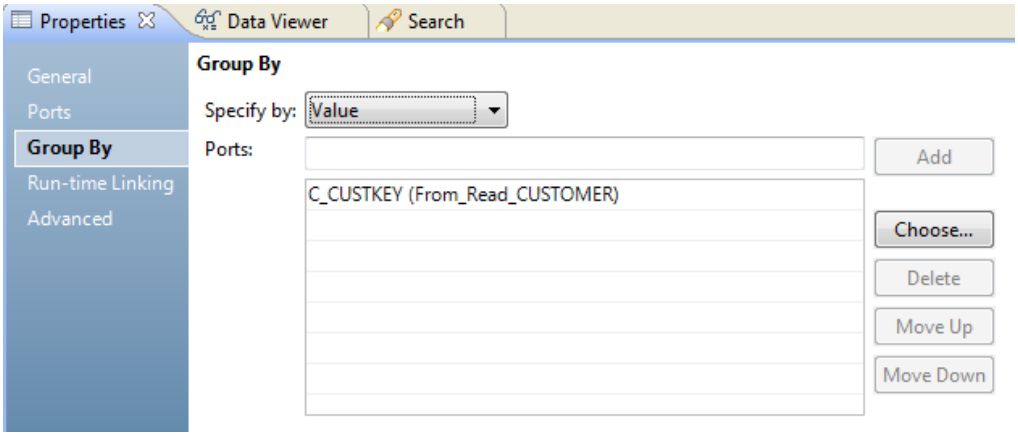
7. Replace the existing expression in the editor with the following expression: `SUM(O_TOTALPRICE_DTL)`



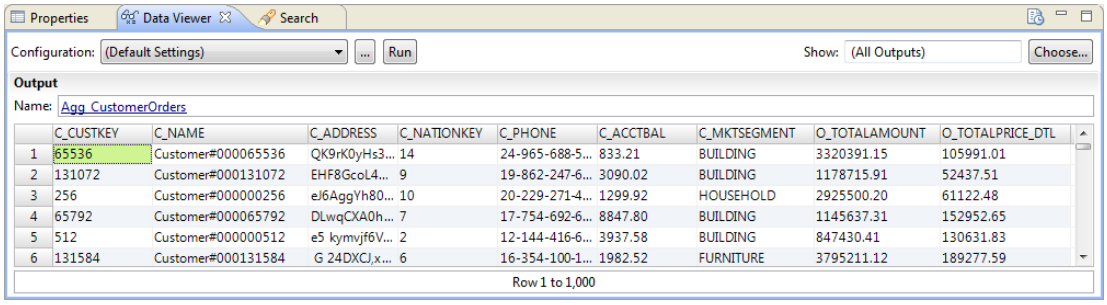
8. Click **Validate** to validate the expression.
9. Click **OK**.
10. Click **OK** to exit the **Aggregator Expression** editor.
11. In the **Properties** view, click the **Group By** tab.
12. Specify the group by port to aggregate the total price by market segment as follows:
    a. Make sure that **Value** from the **Specify by** list is selected.
    b. Click **Choose**.

       The **Ports** dialog box appears.
    c. Select the checkbox next to C_CUSTKEY and click **OK**.

       The following image shows the selected group by port:



You can preview the Aggregator transformation data to make sure that it contains the expected results. In the mapping editor, right-click the Aggregator transformation and select **Run Data Viewer**. The data calculated by the transformation appears in the **Data Viewer** view.



Step 4. Configure the Write Transformation

Add a Write transformation and configure the Write transformation to fetch column changes directly from the target at run time.

1. Add the `CUSTOMERSUMMARY` relational data object as the Write transformation.

   The Write transformation appears in the editor as Write_CUSTOMERSUMMARY.
2. Verify that the Write transformation is configured to automatically re-import metadata changes.
   a. In the **Properties** view, click the **General** tab.
   b. Make sure that **Synchronize input ports** is chosen.
3. Configure the Write transformation to get columns directly from the target table at run time.
   a. In the **Properties** view, click the **Data Object** tab.
   b. Select **At run time, get data object columns from the data source**.

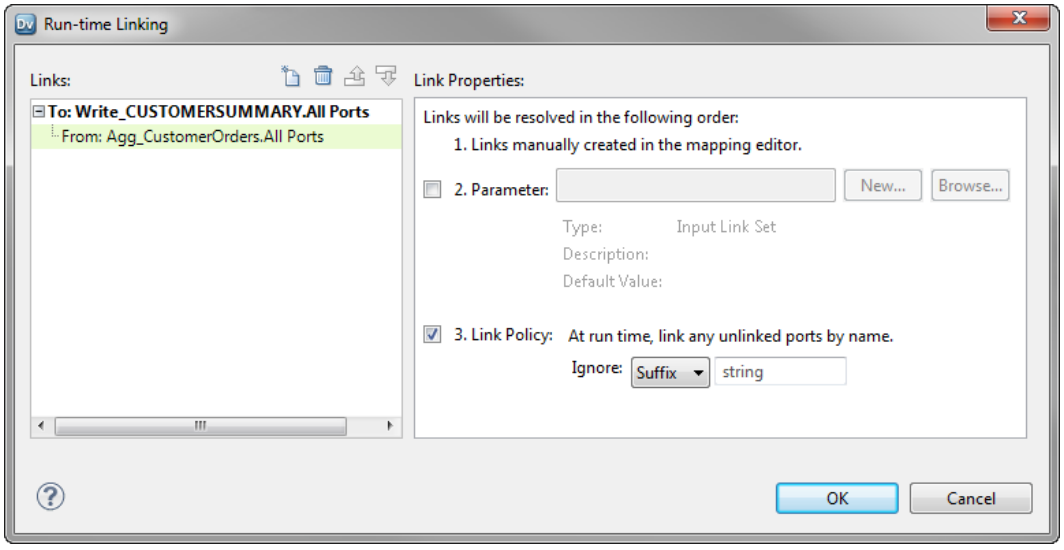
Step 5. Create and Configure a Run-time Link

Create a run-time link to the Write transformation and configure a link policy to establish and resolve links by port names at run time.

1. Press Ctrl and drag the All Ports group from the Aggregator transformation to the All Ports group of the Write transformation.

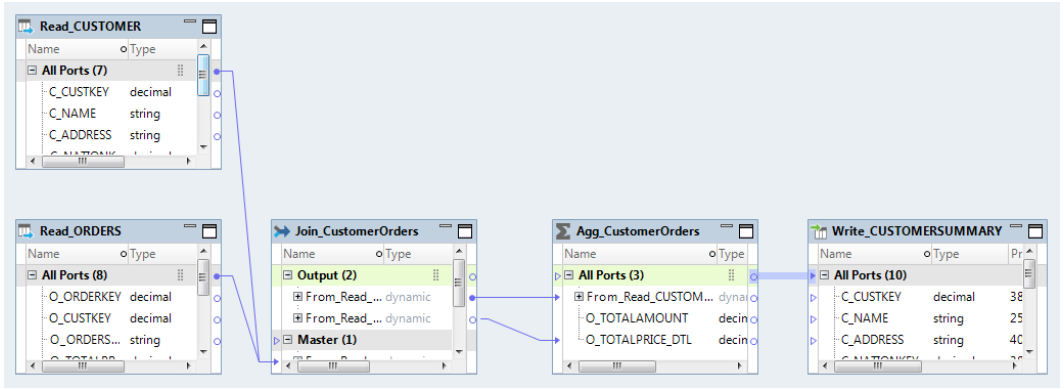
   The **Run-time Linking** dialog box appears.

2. Verify that **Link Policy** in the Link Properties area is selected to automatically link ports by name at run time.



3. Click **OK**.

   The Developer tool creates a run-time link between the Aggregator transformation and the Write transformation.



Step 6. Validate and Run the Mapping

Validate and run the mapping. Preview the data in the target data object to verify the result.

1. In the mapping editor, click **Edit** > **Validate**.
2. When the mapping is valid, click **File** > **Save** to save the mapping.
3. Click **Run** > **Mapping**.

   The **Run Mapping** window displays the progress of the mapping run. The mapping runs and writes the output to the target file.
4. In the **Object Explorer** view, locate the `CUSTOMERSUMMARY` data object in your project and double click the data object.

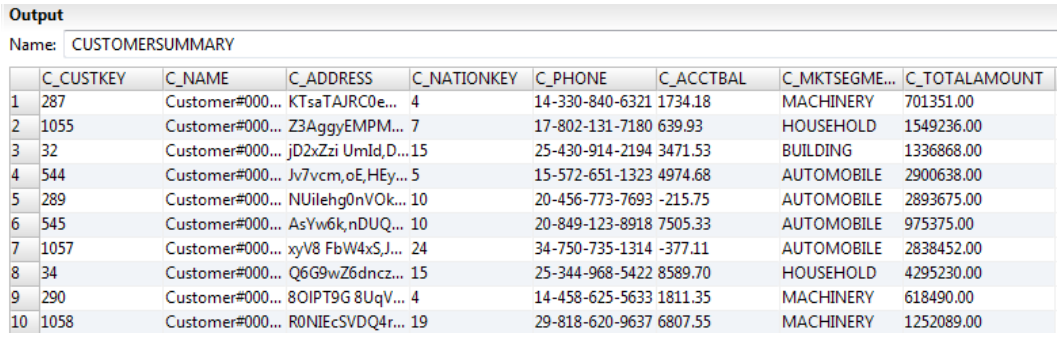
   The data object opens in the editor.
5. Click **Window** > **Show View** > **Data Viewer**.

   The **Data Viewer** view appears.

6. In the **Data Viewer** view, click **Run**.

   The **Data Viewer** view runs and displays the data.

   In this example, the C_TOTALAMOUNT column displays the aggregated total price of customer orders.



Output

Name: CUSTOMERSUMMARY

	C_CUSTKEY	C_NAME	C_ADDRESS	C_NATIONKEY	C_PHONE	C_ACCTBAL	C_MKTSEGME...	C_TOTALAMOUNT
1	287	Customer#000...	KTsaTAJRC0e...	4	14-330-840-6321	1734.18	MACHINERY	701351.00
2	1055	Customer#000...	Z3AggyEMPM...	7	17-802-131-7180	639.93	HOUSEHOLD	1549236.00
3	32	Customer#000...	jD2xZzi UmId,D...	15	25-430-914-2194	3471.53	BUILDING	1336868.00
4	544	Customer#000...	Jv7vcm,oE,HEy...	5	15-572-651-1323	4974.68	AUTOMOBILE	2900638.00
5	289	Customer#000...	NUilehg0nVOk...	10	20-456-773-7693	-215.75	AUTOMOBILE	2893675.00
6	545	Customer#000...	AsYw6k,nDUQ...	10	20-849-123-8918	7505.33	AUTOMOBILE	975375.00
7	1057	Customer#000...	xyV8 FbW4xS,J...	24	34-750-735-1314	-377.11	AUTOMOBILE	2838452.00
8	34	Customer#000...	Q6G9wZ6dncz...	15	25-344-968-5422	8589.70	HOUSEHOLD	4295230.00
9	290	Customer#000...	8OIPT9G 8UqV...	4	14-458-625-5633	1811.35	MACHINERY	618490.00
10	1058	Customer#000...	R0NIEcSVDQ4r...	19	29-818-620-9637	6807.55	MACHINERY	1252089.00

Step 7. Run the Mapping after Changes to the Source Schema

The departments that provide the customer data table and customer order data table add a new column Comments to the tables. View the column changes in the dynamic mapping and validate and rerun the mapping. You can preview the data in the target data object to verify the updated result.

The following table lists the columns and metadata for the updated CUSTOMER table with the new C_COMMENT column:

Name	Native Type	Precision	Scale
C_CUSTKEY	number(p,s)	38	0
C_NAME	varchar2	25	0
C_ADDRESS	varchar2	40	0
C_NATIONKEY	number(p,s)	38	0
C_PHONE	varchar2	15	0
C_ACCTBAL	number(p,s)	10	2
C_MKTSEGMENT	varchar2	10	0
C_COMMENT	varchar2	117	0

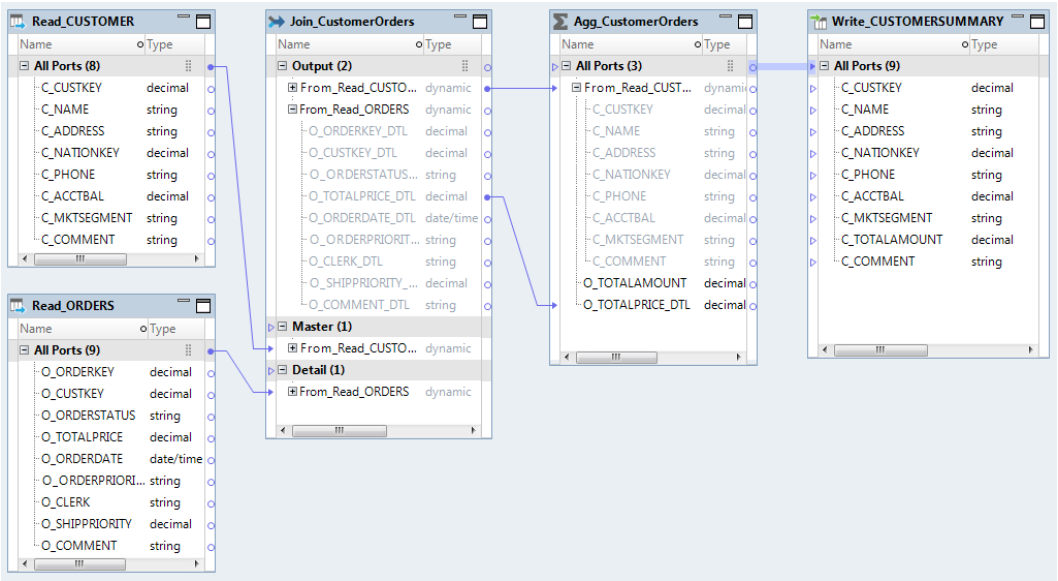
The following table lists the columns and metadata for the updated ORDERS table with the new O_COMMENT column:

Name	Native Type	Precision	Scale
O_ORDERKEY	number(p,s)	38	0
O_CUSTKEY	number(p,s)	38	0

Name	Native Type	Precision	Scale
O_ORDERSTATUS	varchar2	1	0
O_TOTALPRICE	number(p,s)	10	2
O_ORDERDATE	date	19	0
O_ORDERPRIORITY	varchar2	15	0
O_CLERK	varchar2	15	0
O_SHIPPRIOIRITY	number(p,s)	30	0
O_COMMENT	varchar2	79	0

1. In the mapping editor, view the changes to the mapping.

   The Read and Write transformations automatically reflect the new columns. The dynamic ports in the Joiner and Aggregator transformations automatically have the new columns C_COMMENT and O_COMMENT from the respective Read transformations.



2. To validate the changed mapping, click **Edit** > **Validate**.
3. When the mapping is valid, click **File** > **Save** to save the mapping.
4. Click **Run** > **Mapping**.

   The **Run Mapping** window displays the progress of the mapping run. The mapping runs and writes the output to the target file.
5. In the **Object Explorer** view, locate the `CUSTOMERSUMMARY` data object in your project and double click the data object.

   The data object opens in the editor.
6. Click **Window** > **Show View** > **Data Viewer**.

   The **Data Viewer** view appears.

7. In the **Data Viewer** view, click **Run**.

   The **Data Viewer** view runs and displays the data.

8. Verify that the mapping shows the expected results after the source schema has changed.

   The C_TOTALAMOUNT column displays the aggregated total price of customer orders.

Author

Indra Sivakumar
Staff Technical Writer

Acknowledgements

The author would like to acknowledge Lalitha Sundaramurthy and Indranil Roy for their contributions to this article.